[예제 10-1] ex10-01.c

```
#include <unistd.h>
#include <signal.h>

void handler(int signum);
int flag = 5;

main()
{
        struct sigaction act;
        sigset_t set;

        sigemptyset(&(act.sa_mask));
        sigaddset(&(act.sa_mask), SIGALRM);
        sigaddset(&(act.sa_mask), SIGINT);
        sigaddset(&(act.sa_mask), SIGUSR1);

        act.sa_handler = handler;
        sigaction(SIGALRM, &act, NULL);
        sigaction(SIGINT, &act, NULL);
        sigaction(SIGUSR1, &act, NULL);

        printf("call raise(SIGUSR1) before blocking\n");
        raise(SIGUSR1);

        sigemptyset(&set);
        sigaddset(&set, SIGUSR1);
        sigprocmask(SIG_SETMASK, &set, NULL);

        while(flag)
        {
                printf("input SIGINT [%d]\n", flag);
                sleep(1);
        }

        printf("call kill(getpid(), SIGUSR1) after blocking\n");
        kill(getpid(), SIGUSR1);

        printf("sleep by pause.. zzZZ\n");
```

```
        printf("pause return %d\n", pause());

        printf("2 seconds sleeping..zzZ\n");
        alarm(2);
        pause();
}
void handler(int signum)
{
        flag--;

        switch(signum) {
        case SIGINT:
                printf("SIGINT(%d)\n", signum);
                break;
        case SIGALRM:
                printf("SIGALRM(%d)\n", signum);
                break;
        case SIGUSR1:
                printf("SIGUSR1(%d)\n", signum);
                break;
        default:
                printf("signal(%d)\n", signum);
        }
}
```

[예제 10-2] ex10-02.c

```
#include <signal.h>
#include <unistd.h>

main()
{
        sigset_t set;
        int result;

        sigemptyset(&set);
        result = sigismember(&set, SIGALRM);
        printf("SIGALRM is %s a member\n", result ? "" : "not");
        sigaddset(&set, SIGALRM);
        result = sigismember(&set, SIGALRM);
        printf("SIGALRM is %s a member\n", result ? "" : "not");

        sigfillset(&set);
        result = sigismember(&set, SIGCHLD);
        printf("SIGCHLD is %s a member\n", result ? "" : "not");
        sigdelset(&set, SIGCHLD);
        result = sigismember(&set, SIGCHLD);
        printf("SIGCHLD is %s a member\n", result ? "" : "not");
}
```

[예제 10-3] ex10-03.c

```
#include <signal.h>
#include <unistd.h>

int num = 0;

main()
{
        static struct sigaction act;

        void int_handle(int);

        act.sa_handler = int_handle;
        sigfillset(&(act.sa_mask));
        sigaction(SIGINT, &act, NULL);

        while(1)
        {
                printf("i'm sleepy..\n");
                sleep(1);
                if(num >= 3)
                        exit(0);
        }
}

void int_handle(int signum)
{
        printf("SIGINT:%d\n", signum);
        printf("int_handle called %d times\n", ++num);
}
```

[예제 10-4] ex10-04.c

```
#include <signal.h>
#include <unistd.h>

int num = 0;

main()
{
        static struct sigaction act;

        void int_handle(int);

        act.sa_handler = int_handle;
        sigfillset(&(act.sa_mask));
        sigaction(SIGINT, &act, NULL);

        while(1)
        {
                printf("i'm sleepy..\n");
                sleep(1);
                if(num >= 2)
                {
                        act.sa_handler = SIG_DFL;
                        sigaction(SIGINT, &act, NULL);
                }
        }
}

void int_handle(int signum)
{
        printf("SIGINT:%d\n", signum);
        printf("int_handle called %d times\n", ++num);
}
```

[예제 10-5] ex10-05.c

```
#include <unistd.h>
#include <signal.h>

main()
{
        sigset_t set;
        int count = 3;

        sigemptyset(&set);
        sigaddset(&set, SIGINT);

        sigprocmask(SIG_BLOCK, &set, NULL);

        while(count)
        {
                printf("don't disturb me (%d)\n", count--);
                sleep(1);
        }

        sigprocmask(SIG_UNBLOCK, &set, NULL);

        printf("you did not disturb me!!\n");
}
```

[예제 10-6] ex10-06.c

```
#include <unistd.h>
#include <signal.h>
#include <sys/types.h>

main()
{
        pid_t pid;
        int count = 5;

        if((pid = fork()) > 0)
        {
                sleep(2);
                kill(pid, SIGINT);
                raise(SIGINT);
                printf("[parent] bye!\n");
        }
        else if(pid == 0)
        {
                while(count)
                {
                        printf("[childe] count is %d\n", count--);
                        sleep(1);
                }
        }
        else
                printf("fail to fork\n");
}
```

[예제 10-7] ex10-07.c

```
#include <unistd.h>
#include <signal.h>

void timeover(int signum)
{
        printf("\n\ntime over!!\n\n");
        exit(0);
}

main()
{
        char buf[1024];
        char *alpha = "abcdefghijklmnopqrstuvwxyz";

        int timelimit;
        struct sigaction act;

        act.sa_handler = timeover;
        sigaction(SIGALRM, &act, NULL);

        printf("input timelimit (sec)..\n");
        scanf("%d", &timelimit);

        alarm(timelimit);

        printf("START!!\n > ");
        scanf("%s", buf);

        if(!strcmp(buf, alpha))
                printf("well done.. you succeed!\n");
        else
                printf("sorry.. you fail!\n");
}
```

[예제 10-8] ex10-08.c

```
#include <unistd.h>
#include <signal.h>

main()
{
        printf("pause return %d\n", pause());
}
```

[예제 10-9] ex10-09.c

```
#include <unistd.h>
#include <signal.h>

void handler(int signum);

main()
{
        struct sigaction act;

        sigfillset(&(act.sa_mask));
        act.sa_handler = handler;

        sigaction(SIGINT, &act, NULL);

        printf("pause return %d\n", pause());
}

void handler(int signum)
{
        printf("\nSIGINT catched\n\n");
}
```